

Sage: guida rapida

William Stein (basato su P. Jipsen, trad. F. Zanini)
GNU Free Document License, estendibile per usi specifici

Notebook

Sage Notebook

Version 3.4

Sage Quickref

last edited on March 26, 2009 10:04 AM by admin

File... Action... Data... sage Typeset Print Worksheet Edit Text Undo Share Publish

e^(2*pi) + 2/3

e^{2π} + ²/₃

Calcola cella: `<maiusc-invio>`

Calcola creando una nuova cella: `<alt-invio>`

Dividi cella: `<control-; >`

Unisci celle: `<control-backspace>`

Inserisci cella matematica: clicca la riga blu tra due celle

Inserisci cella text/HTML: maiusc-clicca la riga blu

Elimina cella: cancella i contenuti e poi backspace

Linea di comando

`com<tab>` completa *comando*

`*bar*?<tab>` elenca comandi contenenti “bar”

`comando?<tab>` mostra documentazione

`comando??<tab>` mostra sorgente

`a.<tab>` mostra metodi dell’oggetto `a` (di pi: `dir(a)`)

`a._<tab>` mostra metodi nascosti dell’oggetto `a`

`search_doc("stringa o regexp")` ricerca nella doc.

`search_src("stringa o regexp")` ricerca codice sorgente

`_` l’output precedente

Numeri

Interi: **Z** = ZZ es. -2 -1 0 1 10¹⁰⁰

Razionali: **Q** = QQ es. 1/2 1/1000 314/100 -2/1

Reali: **R** ≈ RR es. .5 0.001 3.14 1.23e10000

Complessi: **C** ≈ CC es. CC(1,1) CC(2.5,-3)

Precisione doppia: RDF and CDF es. CDF(2.1,3)

Mod *n*: **Z**/*n***Z** = Zmod es. Mod(2,3) Zmod(3)(2)

Campi finiti: **F**_{*q*} = GF es. GF(3)(2) GF(9,"a").0

Polinomi: *R*[*x,y*] es. S.<*x,y*>=QQ[] x+2*y³

Serie: *R*[[*t*]] es. S.<*t*>=QQ[] 1/2+2*t+0(t²)

Numeri *p*-adici: **Z**_{*p*} ≈ Zp, **Q**_{*p*} ≈ Qp es. 2+3*5+0(5²)

Chiusura algebrica: **Q**¯ = QQbar es. QQbar(2^(1/5))

Aritmetica degli intervalli: RIF es. RIF((1,1.001))

Campo di numeri: R.<*x*>=QQ[] ;K.<*a*>=NumberField(x³)

Aritmetica

$ab = a*b$ $\frac{a}{b} = a/b$ $a^b = a^b$ $\sqrt{x} = \text{sqrt}(x)$
 $\sqrt[n]{x} = x^(1/n)$ $|x| = \text{abs}(x)$ $\log_b(x) = \text{log}(x,b)$

Somme: $\sum_{i=k}^n f(i) = \text{sum}(f(i) \text{ for } i \text{ in } (k..n))$

Prodotti: $\prod_{i=k}^n f(i) = \text{prod}(f(i) \text{ for } i \text{ in } (k..n))$

Costanti e funzioni

Costanti: $\pi = \text{pi}$ $e = \text{e}$ $i = \text{i}$ $\infty = \text{oo}$
 $\phi = \text{golden_ratio}$ $\gamma = \text{euler_gamma}$

Approssima: `pi.n(digits=18)` = 3.14159265358979324

Funzioni: `sin cos tan sec csc cot sinh cosh tanh sech csch coth log ln exp ...`

Funzioni Python: `def f(x): return x^2`

Funzioni interattive

Metti @interact prima della funzione

@interact

`def f(n=[0..4], s=(1..5), c=Color("red")):`
 `var("x");show(plot(sin(n*x^s),-pi,pi,color=c))`

Espressioni simboliche

Definisci nuove variabili simboliche: `var("t u v y z")`

Funzioni simboliche: es. $f(x) = x^2$ `f(x)=x^2`

Relazioni: `f==g` `f<=g` `f>=g` `f<g` `f>g`

Risolvi $f = g$: `solve(f(x)==g(x), x)`
 `solve([f(x,y)==0, g(x,y)==0], x,y)`

`factor(...)` `expand(...)` `(...).simplify...`

`find_root(f(x), a, b)` trova $x \in [a,b]$ s.t. $f(x) \approx 0$

Analisi

$\lim_{x \rightarrow a} f(x) = \text{limit}(f(x), x=a)$

$\frac{d}{dx}(f(x)) = \text{diff}(f(x), x)$

$\frac{\partial}{\partial x}(f(x,y)) = \text{diff}(f(x,y), x)$

`diff = differentiate = derivative`

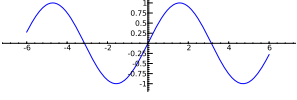
$\int f(x)dx = \text{integral}(f(x), x)$

$\int_a^b f(x)dx = \text{integral}(f(x), x,a,b)$

$\int_a^b f(x)dx \approx \text{numerical_integral}(f(x), a,b)$

Polinomio di Taylor, grado *n* in *a*: `taylor(f(x),x,a,n)`

Grafici 2D



`line([(x1,y1),...,(xn,yn)],opzioni)`

`polygon([(x1,y1),...,(xn,yn)],opzioni)`

`circle((x,y),r,opzioni)`

`text(txt,(x,y),opzioni)`

opzioni come in `plot.options`, es. `thickness=pixel`,
`rgbcolor=(r,g,b)`, `hue=h` dove $0 \leq r,b,g,h \leq 1$

`show(grafico, opzioni)`

`figsize=[w,h]` per cambiare le dimensioni

`aspect_ratio=numero` per cambiare le proporzioni

`plot(f(x),(x,xmin,xmax),options)`

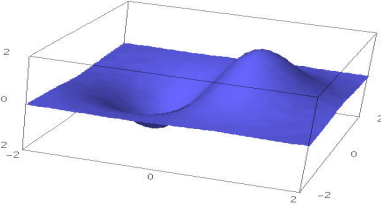
`parametric_plot((f(t),g(t)),(t,tmin,tmax),options)`

`polar_plot(f(t),(t,tmin,tmax),opzioni)`

combina: `circle((1,1),1)+line([(0,0),(2,2)])`

`animate(elenco di grafici, opzioni).show(delay=20)`

Grafici 3D



`line3d([(x1,y1,z1),...,(xn,yn,zn)],opzioni)`

`sphere((x,y,z),r,opzioni)`

`text3d(txt, (x,y,z), opzioni)`

`tetrahedron((x,y,z),dimensione,opzioni)`

`cube((x,y,z),dimensione,opzioni)`

`octahedron((x,y,z),dimensione,opzioni)`

`dodecahedron((x,y,z),dimensione,opzioni)`

`icosahedron((x,y,z),dimensione,opzioni)`

`plot3d(f(x,y),(x,xb,xe), (y,yb,ye),opzioni)`

`parametric_plot3d((f,g,h),(t,tb,te),opzioni)`

`parametric_plot3d((f(u,v),g(u,v),h(u,v)), (u,ub,ue),(v,vb,ve),opzioni)`

opzioni: `aspect_ratio=[1,1,1]`, `color=red`
`opacity=0.5`, `figsize=6`, `viewer=tachyon`

Matematica discreta

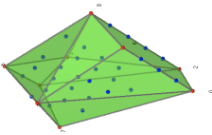
$\lfloor x \rfloor = \text{floor}(x)$ $\lceil x \rceil = \text{ceil}(x)$
Resto di n diviso per $k = n\%k$ $k|n$ sse $n\%k==0$
 $n! = \text{factorial}(n)$ $\binom{x}{m} = \text{binomial}(x,m)$
 $\phi(n) = \text{euler_phi}(n)$
Stringhe: es. $s = \text{"Ciao"} = \text{"Ci"} + \text{'ao'}$
 $s[0]=C$ $s[-1]=o$ $s[1:2]=ia$ $s[2:]=ao$
Elenchi: es. $[1, \text{"Ciao"}, x] = [] + [1, \text{"Ciao"}] + [x]$
Tuple: es. $(1, \text{"Ciao"}, x)$ (immutabile)
Insiemi: es. $\{1, 2, 1, a\} = \text{Set}([1, 2, 1, \text{"a"}])$
Comprensione elenchi \approx notazione costruttore insiemi, es.
 $\{f(x) : x \in X, x > 0\} = \text{Set}([f(x) \text{ for } x \text{ in } X \text{ if } x > 0])$

Teoria dei grafi



Grafo: $G = \text{Graph}(\{0:[1,2,3], 2:[4]\})$
Grafo orientato: $\text{DiGraph}(\text{dictionary})$
Famiglie di grafici: $\text{graphs.}(\text{tab})$
Invarianti: $G.\text{chromatic_polynomial}()$, $G.\text{is_planar}()$
Cammini: $G.\text{shortest_path}()$
Visualizza: $G.\text{plot}()$, $G.\text{plot3d}()$
Automorfismi: $G.\text{automorphism_group}()$,
 $G1.\text{is_isomorphic}(G2)$, $G1.\text{is_subgraph}(G2)$

Calcolo combinatorio



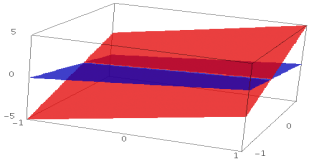
Sequenze di interi: $\text{sloane.find}(\text{list})$, $\text{sloane.}(\text{tab})$
Partizioni: $P = \text{Partitions}(n)$ $P.\text{count}()$
Combinazioni: $C = \text{Combinations}(\text{list})$ $C.\text{list}()$
Prodotto cartesiano: $\text{CartesianProduct}(P,C)$
Tabelle: $\text{Tableau}([[1,2,3], [4,5]])$
Parole: $W = \text{Words}(abc)$; $W(aabca)$
Ordinamenti parziali: $\text{Poset}([[1,2], [4], [3], [4], []])$
Sistemi di radici: $\text{RootSystem}([A,3])$

Cristalli: $\text{CrystalOfTableaux}([A,3], \text{shape}=[3,2])$
Politopi reticolari: $A = \text{random_matrix}(ZZ, 3, 6, x=7)$
 $L = \text{LatticePolytope}(A)$ $L.\text{npoints}()$ $L.\text{plot3d}()$

Algebra di matrici

$\begin{pmatrix} 1 \\ 2 \end{pmatrix} = \text{vector}([1,2])$
 $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \text{matrix}(QQ, [[1,2], [3,4]], \text{sparse=False})$
 $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} = \text{matrix}(QQ, 2, 3, [1,2,3, 4,5,6])$
 $\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} = \text{det}(\text{matrix}(QQ, [[1,2], [3,4]]))$
 $Av = A*v$ $A^{-1} = A^{-1}$ $A^t = A.\text{transpose}()$
Risolvi $Ax = v$: $A \backslash v$ or $A.\text{solve_right}(v)$
Risolvi $xA = v$: $A.\text{solve_left}(v)$
Forma triangolare superiore: $A.\text{echelon_form}()$
Rango e nullit: $A.\text{rank}()$ $A.\text{nullity}()$
Forma di Hessenberg: $A.\text{hessenberg_form}()$
Polinomio caratteristico: $A.\text{charpoly}()$
Autovalori: $A.\text{eigenvalues}()$
Autovettori: $A.\text{eigenvectors_right}()$ (also left)
Gram-Schmidt: $A.\text{gram_schmidt}()$
Visualizza: $A.\text{plot}()$
Riduzione LLL: $\text{matrix}(ZZ, \dots).\text{LLL}()$
Forma di Hermite: $\text{matrix}(ZZ, \dots).\text{hermite_form}()$

Algebra lineare



Spazio vettoriale $K^n = K^n$ e.g. QQ^3 RR^2 CC^4
Sottospazio: $\text{span}(\text{vectors}, \text{field})$, es.
 $\text{span}([[1,2,3], [2,3,5]], QQ)$
Nucleo: $A.\text{right_kernel}()$ (anche sinistro)
Somma e intersezione: $V + W$ e $V.\text{intersection}(W)$
Base: $V.\text{basis}()$
Matrice di base: $V.\text{basis_matrix}()$
Restringi matrice a un sottospazio: $A.\text{restrict}(V)$
Vettore scritto su una base: $V.\text{coordinates}(\text{vector})$

Matematica numerica

Pacchetti: $\text{import numpy, scipy, cvxopt}$
Minimizzazione: $\text{var}(\text{"x y z"})$
 $\text{minimize}(x^2 + x*y^3 + (1-z)^2 - 1, [1,1,1])$

Teoria dei numeri

Primi: $\text{prime_range}(n,m)$, is_prime , next_prime
Fattorizza: $\text{factor}(n)$, $\text{qsieve}(n)$, $\text{ecm.factor}(n)$
Simbolo di Kronecker: $\left(\frac{a}{b}\right) = \text{kronecker_symbol}(a,b)$
Frazioni continue: $\text{continued_fraction}(x)$
Numeri di Bernoulli: $\text{bernoulli}(n)$, $\text{bernoulli_mod_p}(p)$
Curve ellittiche: $\text{EllipticCurve}([a_1, a_2, a_3, a_4, a_6])$
Caratteri di Dirichlet: $\text{DirichletGroup}(N)$
Forme modulari: $\text{ModularForms}(\text{level}, \text{weight})$
Simboli modulari: $\text{ModularSymbols}(\text{level}, \text{weight}, \text{sign})$
Moduli di Brandt: $\text{BrandtModule}(\text{level}, \text{weight})$
Variet modulari Abelian: $J0(N)$, $J1(N)$

Teoria dei gruppi

$G = \text{PermutationGroup}([[(1,2,3), (4,5)], [(3,4)]])$
 $\text{SymmetricGroup}(n)$, $\text{AlternatingGroup}(n)$
Gruppi abeliani: $\text{AbelianGroup}([3,15])$
Gruppi di matrici: GL , SL , Sp , SU , GU , SO , GO
Funzioni: $G.\text{syLOW_subgroup}(p)$, $G.\text{character_table}()$,
 $G.\text{normal_subgroups}()$, $G.\text{cayley_graph}()$

Anelli non commutativi

Quaternioni: $Q.<i,j,k> = \text{QuaternionAlgebra}(a,b)$
Algebra libera: $R.<a,b,c> = \text{FreeAlgebra}(QQ, 3)$

Moduli Python

$\text{import nome_del_modulo}$
 $\text{module_name.}(\text{tab})$ e $\text{help}(\text{module_name})$

Personalizzazione e debugging

time comando : mostra informazioni di timing
 $\text{timeit}(\text{"comando"})$: misura il tempo del comando
 $t = \text{cputime}()$; $\text{cputime}(t)$: tempo CPU trascorso
 $t = \text{walltime}()$; $\text{walltime}(t)$: tempo reale trascorso
 $\%pdb$: attiva debugger interattivo (solo linea di comando)
 $\%prun$ comando: personalizza comando (solo ldc)